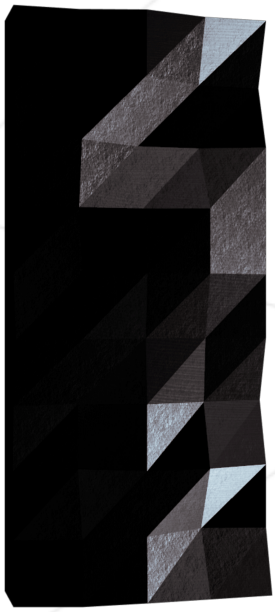


Datascience X Logistic Regression

Harry Potter et le datascientist

Résumé: Codez un classifieur et sauvez Poudlard !

Table des matières

I Préambule **2**

II Introduction **3**

III Objectifs **4**

IV Consignes générales **5**

V Partie obligatoire **6**
V.1 Data Analysis 6
V.2 Data Visualization 7
V.2.1 Histogram 7
V.2.2 Scatter plot 7
V.2.3 Pair plot 7
V.3 Logistic Regression 8

VI Partie bonus **9**

VII Rendu et peer-évaluation **10**

VIII Annexes **11**
VIII.1 Mathematics 11
VIII.2 Data visualization examples 12

Chapitre I

Préambule

Voici ce que dit Wikipédia de Yann Le Cun, notre *Pape* de l'IA :

Yann Le Cun, né le 8 juillet 1960 à Paris, est un chercheur en intelligence artificielle et vision artificielle (robotique). Il est considéré comme l'un des inventeurs du deep learning.

Yann Le Cun est diplômé de l'ESIEE Paris en 1983, il part ensuite à l'Université Pierre-et-Marie-Curie effectuer un DEA puis un doctorat qu'il obtient en 1987. Il s'oriente rapidement vers la recherche sur l'apprentissage automatique et il propose pendant sa thèse une variante de l'algorithme de rétropropagation du gradient, qui permet depuis le début des années 1980 l'apprentissage des réseaux de neurones. Il réalise son post-doctorat au sein de l'équipe de Geoffrey Hinton.

Yann Le Cun travaille depuis les années 1980 sur l'apprentissage automatique (machine learning) et l'apprentissage profond (deep learning) : la capacité d'un ordinateur à reconnaître des représentations (images, textes, vidéos, sons) à force de les lui montrer, de très nombreuses fois.

En 1987, Yann Le Cun rejoint l'Université de Toronto et en 1988 les laboratoires AT & T, pour lesquels il développe des méthodes d'apprentissage supervisé.

Yann Le Cun s'intéresse ensuite à la conception des algorithmes de compression du format d'archivage DjVu puis à la reconnaissance automatique de chèque bancaires.

Yann Le Cun est professeur à l'université de New York où il a créé la Center for Data Sciences. Il travaille notamment au développement technologique des voitures autonomes.

Le 9 décembre 2013, Yann Le Cun est invité par Mark Zuckerberg à rejoindre Facebook pour créer et diriger le laboratoire d'intelligence artificielle FAIR (« Facebook Artificial Intelligence Research ») à New York, Menlo Park et depuis 2015 à Paris, notamment pour travailler sur la reconnaissance d'images et de vidéos. Il avait précédemment refusé une proposition similaire de la part de Google.

En 2016, Yann Le Cun est le titulaire pour l'année de la chaire « Informatique et sciences numériques » du Collège de France.

Chapitre II

Introduction

Horreur ! Depuis sa création, la célèbre école de sorcier, Poudlard, n'avait jamais connu pareille offense. Les forces du mal ont ensorcelé le Choixpeau magique. Ce dernier ne répond plus, comme hébêté, il est incapable de remplir son rôle en répartissant les élèves dans les fameuses maisons.

La rentrée approche. Le professeur McGonagall a su prendre des mesures à la hauteur de la situation, impossible pour Poudlard de ne pas accueillir de nouveaux étudiants. . .Elle décide de faire appel à vous, un moldu "datascientist" et capable de faire des miracles avec cet outil dont tous les moldus connaissent l'usage : un "ordinateur".

Malgré la réticence de beaucoup de sorciers, la directrice de l'école vous accueille dans son bureau pour vous expliquer la situation. Si vous êtes ici c'est parce que son contact lui a révélé que vous êtes capable de recréer un Choixpeau magique à l'aide de vos seuls outils de moldus. Vous acquiescez. Vous expliquez que pour que la "magie" opère, vous devez récupérer des données sur les élèves. McGonagall vous remet alors un grimoire poussiéreux. Heureusement pour vous, un simple "Digitalis !" et le livre se change en clef USB.

Chapitre III

Objectifs

Dans ce projet *DataScience x Logistic Regression*, vous allez poursuivre votre exploration du *Machine Learning* en ajoutant divers outils à votre escarcelle.

L'utilisation du terme *DataScience* dans le titre sera sûrement considérée par certains comme abusive. C'est vrai. Nous ne prétendons pas dans ce sujet vous donner toutes les bases de *DataScience*. Le sujet est vaste. Nous verrons ici des bases qui nous ont paru utiles pour traiter un jeu de données avant de l'envoyer dans un algorithme de *machine learning*.

Vous implémenterez un modèle de classification linéaire, dans la continuité du sujet *linear regression* : une *logistic regression*. Nous vous encourageons d'ailleurs beaucoup à vous créer une bibliothèque de *machine learning* au fur et à mesure que vous avancerez dans la branche.

En somme :

- Vous apprendrez à lire un jeu de données, à le visualiser de différentes manières, à sélectionner et nettoyer vos données.
- Vous mettrez en place une régression logistique qui vous permettra de résoudre des problèmes de classification.

Chapitre IV

Consignes générales

Ce sujet est en langage libre. Nous vous recommandons toutefois de choisir un langage doté d'une librairie permettant d'afficher facilement des graphes et une autre pour le calcul.

Toutes librairies qui feraient le travail à votre place (comme utiliser la fonction `describe` de la librairie *Pandas*) sera considéré comme triche.

Chapitre V

Partie obligatoire

Il est chaudement recommandé d'effectuer les étapes dans l'ordre.

V.1 Data Analysis

Nous allons voir quelques étapes basiques d'exploration des données. Bien entendu, ce ne sont pas les seules techniques disponibles ni la seule marche à suivre. Chaque jeu de données et problème demandent d'être abordé de manière unique. Vous aurez sûrement à trouver d'autres manières de comprendre vos données à l'avenir.

Avant toutes choses, jetez un oeil aux données disponibles. Regardez sous quel format elles sont enregistrées, s'il y a divers types de données, les différentes fourchettes, etc. Il est important de se faire une idée de votre matière première avant de commencer à travailler. Plus vous travaillerez sur des données et plus vous développerez une intuition sur comment vous allez pouvoir vous en servir.

Dans cette partie, le professeur McGonagall vous demande de produire un programme nommé `describe.[extension]`. Ce programme prendra un *dataset* en paramètre. Il devra ni plus ni moins afficher les informations sur toutes les *features* numériques comme l'exemple qui suit :

```
$> describe.[extension] dataset_train.csv
          Feature 1     Feature 2     Feature 3     Feature 4
Count    149.000000    149.000000    149.000000    149.000000
Mean       5.848322      3.051007      3.774497      1.205369
Std        5.906338      3.081445      4.162021      1.424286
Min        4.300000      2.000000      1.000000      0.100000
25%        5.100000      2.800000      1.600000      0.300000
50%        5.800000      3.000000      4.400000      1.300000
75%        6.400000      3.300000      5.100000      1.800000
Max        7.900000      4.400000      6.900000      2.500000
```

Il est interdit d'utiliser des fonctions qui font le café pour vous comme : count, mean, std, min, max, percentile, etc... quelque soit le langage que vous utilisez. Bien entendu, il est aussi interdit d'utiliser la fonction describe de la librairie Pandas ou une fonction qui y ressemble de près ou de loin dans une autre librairie.

V.2 Data Visualization

La visualisation des données est un outil puissant pour le datascientist. Cela vous permet d'acquérir une intuition sur comment les données sont connectées les unes aux autres. Visualiser vos données vous permet aussi de déceler plusieurs défauts.

Dans cette section, il vous est demandé de répondre à une question en créant un script pour chacune des questions qui affichera une visualisation particulière. Il n'y a pas forcèment une seule réponse valable à la question.

V.2.1 Histogram

Faites un script nommé `histogram.[extension]` qui affiche un *histogram* répondant à la question suivante :

Quel cours de Poudlard a une répartition des notes homogènes entre les quatres maisons ?

V.2.2 Scatter plot

Faites un script nommé `scatter_plot.[extension]` qui affiche un *scatter plot* répondant à la question suivante :

Quelles sont les deux *features* qui sont semblables ?

V.2.3 Pair plot

Faites un script nommé `pair_plot.[extension]` qui affiche un *pair plot* ou *scatter plot matrix* (selon la librairie graphique que vous utiliserez).

À partir de cette visualisation, quelles caractéristiques allez-vous utiliser pour entraîner votre prochaine régression logistique ?

V.3 Logistic Regression

Vous arrivez à la dernière partie : coder votre Choixpeau magique. Pour ce faire, il vous est demandé de réaliser un multi-classifieur en utilisant une régression logistique en *one-vs-all*.

Vous devrez rendre deux programmes :

- un premier qui va *train* vos modèles, il se nomme `logreg_train.[extension]`. Il prend en paramètre `dataset_train.csv`. Pour la partie obligatoire, vous devez utilisez la technique du *gradient descent* pour minimiser l'erreur. Le programme génère un fichier contenant les poids qui seront réutilisés pour la prédiction.
- un second qui se nomme `logreg_predict.[extension]`. Il prend en paramètre `dataset_test.csv` et un fichier contenant les poids entraînés au préalable.
  Afin d'évaluer les performances de votre classifieur ce second programme devra génèrer un fichier de prédictions `houses.csv` formatté strictement de la manière suivante :

```
$> cat houses.csv
Index,Hogwarts House
0,Gryffindor
1,Hufflepuff
2,Ravenclaw
3,Hufflepuff
4,Slytherin
5,Ravenclaw
6,Hufflepuff
      [...]
```

Chapitre VI

Partie bonus

Il est possible de faire plein de bonus pour ce sujet. Voici quelques suggestions :

- implémenter plus de champs pour `describe.[extension]`
- implémenter un *stochastic gradient descent*
- implémenter une fonction d'optimisation autre
- ...

Chapitre VII

Rendu et peer-évaluation

Rendez-votre travail sur votre dépot `GiT` comme d'habitude. Seul le travail présent sur votre dépot sera évalué en soutenance.

Au cours de la correction vous serez évalués sur votre rendu (aucune utilisation de fonctions qui font le travail à votre place) et votre capacité à expliquer, justifier et démontrer vos choix.

Votre classifieur sera évalué sur les données présentes dans `dataset_test.csv`. Vos réponses seront évaluées en utilisant accuracy score de la bibliothèque *Scikit-Learn*. Mc Gonagall estime que votre algorithme sera l'égal du Choixpeau que si il a une précision de **98%** minimum.

Il sera aussi important de pouvoir expliquer le fonctionnement des algorithmes de *machine learning*.

Chapitre VIII

Annexes

VIII.1 Mathematics

La régression logistique fonctionne pratiquement comme la régression linéaire. Voici la fonction de coût (ou *loss*) :

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m} y^i \log(h_\theta(x^i)) + (1 - y^i)\log(1 - h_\theta(x^i))$$

Où $h_\theta(x)$ est définit de la manière suivante :

$$h_\theta(x) = g(\theta^T x)$$

Avec :

$$g(z) = \frac{1}{1 + e^{-z}}$$

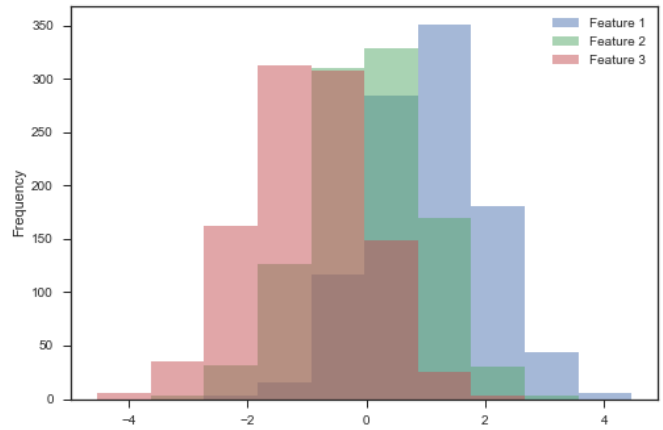
La fonction de coût nous donne la dérivée partielle suivante :

$$\frac{\partial}{\partial \theta_j} J(\theta) = \frac{1}{m}\sum_{i=1}^{m} (h_\theta(x^i) - y^i) x_j^i$$

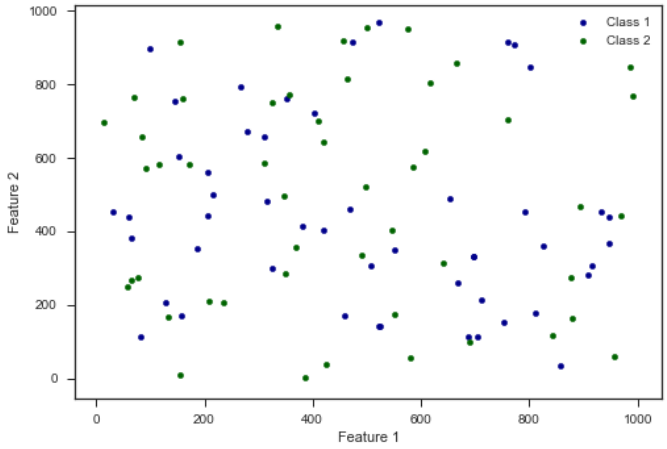
VIII.2 Data visualization examples

Voici des exemples de visualisation de données :

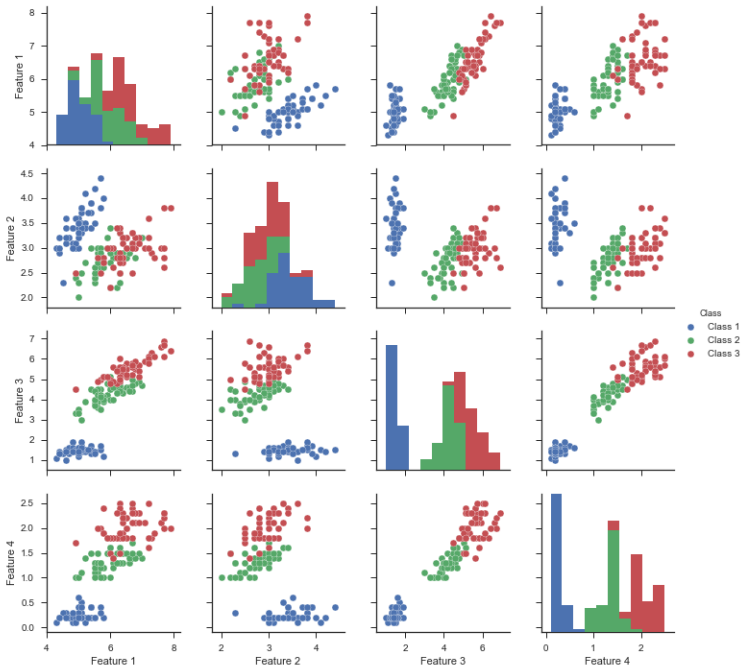
- Histogram



- Scatter plot



- Pair plot


Feature 1
Feature 2
Feature 3
Feature 4
Class
Class 1
Class 2
Class 3